Experimenting with FIASCO for protein structure prediction

F. Campeotto[1,2], A. Dal Palù[3], A. Dovier[1], F. Fioretto[1,2], and E. Pontelli[2]

[1] Dept. Mathematics & Computer Science, University of Udine
[2] Dept. Computer Science, New Mexico State University
[3] Dept. Mathematics, University of Parma

Abstract

Constraint Programming (CP) [6] is a declarative programming methodology that has gained a predominant role in addressing large scale combinatorial and optimization problems. The CP paradigm provides the tools necessary to guide the modeling and resolution of search problems. In particular, it enables the declarative modeling (in terms of variables and constraints) of problems, the ability to rapidly propagate the effects of search decisions, and flexible and efficient procedures to explore the search space of possible solutions.

The focus of this work is the use of constraint-based technology in the investigation of protein structures. We developed a system composed of two main parts: **(1)** a *Graphical User Interface*, to help the user, in particular users coming from Biology, in modeling protein structure studies with CP, and **(2)** a constraint solver targeted at protein structure analysis. Parts of the system have been presented in four previous works. In [1] a first prototype of the graphical interface has been presented, using a preliminary version of the solver later presented in [2] and, still improved, in [5]. This solver focused on one class of constraints targeting the problem of loop closure; our current work provides instead a *comprehensive* constraint system for modeling generic structural protein properties and investigating different types of problems (e.g., structure prediction, studies of flexibility). Moreover the solver makes use, in some parts, of GPU computation following the ideas presented in [4], as explained below.

1. The Graphical User Interface. The user interface is developed in Java and allows the user to impose constraints on protein structures as well as to visualize the results of the solver's execution. The interface allows the user to select the primary structure of a target protein as input and to launch the solver. In turn, the solver computes a preliminary protein's structure using default constraints, according to the sequence of amino acids given in input. The structure is displayed to the user, who can choose to add new constraints (e.g., distance constraints between secondary structure elements) and re-run the solver. This process can be iterated until a desired solution is found.

2. The Solver. The constraint solver is composed by two sub-systems: **(a)** a parallel *Multi-Agents System* [4], and **(b)** the FIASCO constraint solver [5].

(a) applies a multi-agents system (*MAS*) perspective, where concurrent agents explore the folding of different parts of a protein. The strength of the approach lies in the agents' ability to apply different types of knowledge, expressed in the form of declarative constraints, to prune the search space of folding alternatives. Distinct agents are in charge of retrieving, filtering, and coordinating local information about parts of a protein, aiming to reach a global consensus. This system is further enhanced by the use of a *General-Purpose Graphical Processing Unit (GPGPU)* [7] to implement the MAS infrastructure, with significant performance improvements over the sequential implementation—available if the tool is not run on a suitable GPU. The multi-agents system is used to predict the preliminary protein's structure on which the user will impose additional constraints to be satisfied by FIASCO. GPGPUs have been recently used in the implementation of a generic constraint solver [3].
(b) is an efficient C++-based resolution engine that can be used to model and solve a class of problems derived from loop modeling instances. FIASCO provides a uniform and efficient modeling platform for studying different structural properties—that have been, so far, addressed only using significantly distinct methods and tools. The declarative nature of constraint-based methods supports a level of elaboration tolerance that is not offered by other frameworks for protein structure prediction, facilitating the integration of additional knowledge in guiding the studies (e.g., availability of information on secondary structure).

Acknowledgments. We thank Matteo Dusefante for his support for improving the GUI and Federico Fogolari for his advices on several parts of this work. The work is partially supported by GNCS-INdAM 2014 project.

References

1. M. Best, K. Bhattarai, F. Campeotto, A. Dal Palù, H. Dang, A. Dovier, F. Fioretto, F. Fogolari, T. Le, and E. Pontelli. Introducing FIASCO: Fragment-based Interactive Assembly for protein Structure prediction with COnstraints. *Proc. of WCB 2011*, co-located to CP2011, Perugia, September 2011.
2. F. Campeotto, A. Dal Palù, A. Dovier, F. Fioretto, and E. Pontelli. A Filtering Technique for Fragment Assembly- Based Proteins Loop Modeling with Constraints *Proc. of CP*, LNCS 7514, pp. 850–866, Springer, 2012.
3. F. Campeotto, A. Dal Palù, A. Dovier, F. Fioretto, and E. Pontelli. Exploring the Use of GPUs in Constraint Solving. *Proc. of PADL*, LNCS 8324, pp. 152–167, Springer, 2014.
4. F. Campeotto, A. Dovier, and E. Pontelli. Protein structure prediction on GPU: a declarative approach in a multi-agent framework. *Proc. of ICPP*, pp. 474–479, IEEE, 2013.
5. F. Campeotto, A. Dal Palù, A. Dovier, F. Fioretto, and E. Pontelli. A Constraint Solver for Flexible Protein Models. JAIR (Journal of Artificial Intelligence Research) 48:953–1000, 2013.
6. F. Rossi, P. van Beek, and T. Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*, Elsevier, 2006.
7. J. Sanders and E. Kandrot. *CUDA by Example. An Introduction to General-Purpose GPU Programming.* Addison Wesley, 2010.